A BOOSTING APPROACH TO CLUSTERING

D. Frossyniotis, A. Likas and A. Stafylopatis

26– 2002

Preprint, no 26 – 02 / 2002

Department of Computer Science
University of Ioannina
45110 Ioannina, Greece

A Boosting Approach to Clustering

Dimitrios Frossyniotis[1], Aristidis Likas[2], and Andreas Stafylopatis[1]

[1] National Technical University of Athens, Department of Electrical and Computer Engineering, Zographou 157 73, Athens, Greece
{dfros, andreas}@cs.ntua.gr
[2] University of Ioannina, Department of Computer Science, 451 10 Ioannina, Greece
arly@cs.uoi.gr

Abstract. It is widely recognized that the boosting methodology provides superior results for classification problems. In this paper, we propose the Boost-clustering algorithm which constitutes a novel clustering methodology that exploits the general principles of boosting in order to provide a consistent partitioning of the dataset. The Boost-clustering algorithm constitutes a multi-clustering method. At each boosting iteration, a new training set is created using weighted random sampling from the original dataset and a simple clustering algorithm (eg. k-means) is applied to provide a new data partitioning. The final clustering solution is produced by aggregating the multiple clustering results through weighted voting. Experiments on both artificial and real-world data sets indicate that Boost-clustering provides solutions of improved quality.

1 Introduction

Unlike classification problems, there are no established approaches that combine multiple clusterings. This problem is more difficult than designing a multi-classifier system: in the classification case it is straightforward whether a basic classifier (weak learner) performs well with respect to a training point, while in the clustering case this task is difficult since there is a lack of knowledge concerning the label of the cluster in which a training point actually belongs to.

In [1] a multi-clustering fusion method is presented based on combining the results from several runs of a clustering algorithm in order to specify a common partition. Another multi-clustering approach is introduced in [2], where multiple clusterings (using k-means) are exploited to determine a co-association matrix of patterns, which is used to define an appropriate similarity measure that is subsequently used to extract arbitrarily shaped clusters. Model structure selection is sometimes left as a design parameter, while in other instances the selection of the optimal number of clusters is incorporated in the clustering procedure [3, 4] using either local or global cluster validity criteria [5].

The present work, proposes a new cluster ensemble approach based on boosting, whereby multiple clusterings are sequentially constructed to deal with data points which were found hard to be clustered in previous stages. The key feature of this method relies on the general principles of the Boosting classification

algorithm [6] which proceeds by building weak classifiers using patterns that are increasingly difficult to classify. The very good performance of the Boosting method in classification tasks was a motivation to believe that boosting a simple clustering algorithm (weak learner) can lead to a multi-clustering solution with improved performance in terms of robustness and quality of the partitioning. Nevertheless, it must be noted that developing a boosting algorithm for clustering is not a straightforward task, since there exist several issues to be treated as discussed in the next section. The proposed method is general and any type of basic clustering algorithm can be used as the weak learner.

2 The Boost-Clustering Method

We propose a new iterative multiple clustering approach, called *Boost-Clustering*, which iteratively recycles the training examples providing multiple clusterings and resulting in a common partition. At each iteration, a distribution over the training points is computed and a new training set is constructed using random sampling from the original dataset. Then a basic clustering algorithm is applied to partition the new training set. The final clustering solution is produced by aggregating the obtained partitions using weighted voting, where the weight of each partition is a measure of its quality. The algorithm is summarized below.

Algorithm Boost-Clustering

Given: Input sequence of N instances: $(x_1, \ldots, x_N)$, $x_i \in \mathbb{R}^d, i = 1, \ldots, N$, a **basic clustering algorithm**, the number of C clusters to partition the data set and the values of parameters T and δ.

1. Initialize $W_i^0 = 1/N$ for $i = 1, \ldots, N$. Set $t = 1$.
2. Iterate while $t \leq T$
   - Produce a bootstrap replicate of the original data set according to the probability W_i^t for every instance i by resampling with replacement from the original data set.
   - Call the **basic clustering algorithm** H^t, to partition the bootstrap training data.
   - Get the cluster hypothesis $H_i^t = (h_{i,1}^t, h_{i,2}^t, \ldots, h_{i,C}^t)$ for all i, $i = 1, \ldots, N$, where $h_{i,j}$ is the membership degree of instance i to cluster j.
   - Renumber the cluster indexes of H^t according to the highest matching score, given by the fraction of shared instances with the clusters provided from the Boost-Clustering until now.
   - Calculate the pseudoloss:

$$\epsilon_t = \sum_{i=1}^{N} W_i^t (1 - h_{i,\,good}^t + h_{i,\,bad}^t) \tag{1}$$

   - Set $\beta_t = \delta + \epsilon_t$
   - Update distribution W_i:

$$W_i^{t+1} = \frac{W_i^t \beta_t^{(1 - h_{i,\,good}^t + h_{i,\,bad}^t)}}{Z_t} \tag{2}$$

   where Z_t is a normalization constant such that W_i^{t+1} is a distribution, ie., $\sum_{i=1}^{N} W_i^{t+1} = 1$.

- $t := t + 1$

3. Output the final cluster hypothesis:

$$H_i^f = argmax \sum_{j=1}^{t} \left[log\left(\frac{1}{\beta_j}\right) h_{i,k}^j \right] \quad k = 1, \ldots, C \tag{3}$$

It is clear that the approach has been developed following the steps of the boosting algorithm for classification. We assume a given set X of N d-dimensional instances x_i, a basic clustering algorithm (weak learner) and the required number of clusters C. The number T of Boost-Clustering iterations will be considered fixed, although this parameter could be determined automatically using an appropriate stopping criterion (e.g. a validation set). The clustering obtained at iteration t will be denoted as H^t, while H_{ag}^t will denote the aggregate partitioning obtained using clusterings H^τ for $\tau = 1, \ldots, t$. Consequently, for the final partitioning H^f produced by the clustering ensemble it will hold that $H^f = H_{ag}^T$. The basic feature of the method is that at each iteration t a weight W_i^t is computed for each instance x_i such that the higher the weight, the more difficult is for x_i to be clustered. In accordance with the boosting methodology the weight W_i^t constitutes the probability of including x_i in the training set constructed at iteration $t + 1$. At the beginning the weights of all instances are equally initialized, ie. $W_i^0 = 1/N$.

At each iteration $t = 1, \ldots, T$, first a dataset X_t is constructed by sampling from X using the distribution W^t and then a partitioning result H^t is produced using the basic clustering algorithm on the dataset X_t. For each instance x_i we get a cluster hypothesis $H_i^t = (h_{i,1}^t, h_{i,2}^t, \ldots, h_{i,C}^t)$ for all i, $i = 1, \ldots, N$, where $h_{i,j}$ denotes the membership degree of instance i to cluster j (we assume that $\sum_{j=1}^{C} h_{i,j}^t = 1$ for all i). It must be emphasized that, although the basic clustering method may be parametric, the boost-clustering method is non-parametric in the sense that the final partitioning is specified in terms of the membership degrees $h_{i,j}$ and not through the specification of some model parameters (e.g. cluster centers). This fact gives the flexibility to define arbitrarily shaped data partitions and makes necessary the use of non-parametric cluster validity measures as described in the next section.

In the above methodology the most critical issue to be addressed, is how to evaluate the "good" ($h_{i,good}^t$) and "bad" ($h_{i,bad}^t$) clustering of an instance x_i for the partition H^t. In our implementation, we computed $h_{i,\,good}^t$ as the maximum membership degree of x_i to a cluster and $h_{i,\,bad}^t$ as the minimum membership degree to a cluster. Based on these quantities, at each iteration t the pseudoloss ϵ_t is computed using (1).

Then, in analogy with the classification case, the weight distribution W_i^{t+1} for the next iteration is computed using (2), where $\delta \geq 1$ is a predefined constant. Using this formula, we reduce the weight of a well-clustered data point (ie. that belongs to a cluster with a high membership degree) and favour the sampling of badly-clustered data points. Thus, in analogy with the general principle of the Boosting classification algorithm (where specialized classifiers are serially constructed to deal with data points misclassified in previous stages), the Boost-Clustering algorithm clusters in every iteration data points that was hard to be clustered in previous iterations.

A second important issue is related with the *cluster correspondence* problem. This means that in order to define the H^t at iteration t we have to assign an index $l \in \{1, \ldots, C\}$ to each of the C partitions and this indexing must be consistent with those in previous iterations. In particular, we have to decide the one-to-one correspondence between a cluster in partitioning H^t with a cluster in the partition H_{ag}^{t-1}. This correspondence is specified by computing the common patterns between a cluster in H^t

and the clusters in H_{ag}^{t-1}. Then, according to the highest matching score, given by the fraction of common samples, the cluster indexes of H^t are renumbered.

In the proposed method the aggregate clustering result at iteration t is obtained by applying for every instance x_i a weighted voting scheme over the cluster subhypotheses $h_{i,k}^j$, $k = 1, \ldots, C$ using (3), where $log(1/\beta_j)$ is the weight of the contribution of $h_{i,k}^j$ to the aggregate decision. In analogy with the classification case, the weight of a subhypothesis $h_{i,k}^j$ is defined as $log(1/\beta_j)$ so that a greater weight is assigned to subhypotheses with lower error ϵ_t.

3 Experimental Results and Discussion

In the experiments with the Boost-Clustering method, we considered two types of basic clustering algorithms, namely the k-means and the fuzzy c-means. The resulting Boost-Clustering method with k-means as the basic clustering algorithm will be referred to as *Boost-k-means* and with fuzzy c-means *Boost-FCM*, respectively. As mentioned in the previous section, a membership degree $h_{i,j}$ for every instance i to cluster j must be produced using the basic clustering algorithm on the dataset. In our implementation the memberhip degree for both types of basic clustering algorithms, k-means and fuzzy c-means, is given based on the Euclidean distance d:

$$h_{i,j} = \frac{1}{\sum_{k=1}^{C} \frac{d(\vec{x_i}, \vec{\mu}_j)}{d(\vec{x_i}, \vec{\mu}_k)}} \quad i = 1, \ldots, N \quad j = 1, \ldots, C \tag{4}$$

where $\vec{x_i} \in \mathbb{R}^d$ is a training sample and $\vec{\mu} \in \mathbb{R}^d$ corresponds to a cluster center.

In the experimental study, we compare Boost-k-means with the simple k-means algorithm and Boost-FCM with the simple FCM algorithm. To accomplish that, non-parametric cluster-validity measures must be specified as noted in the previous section.

3.1 Non-parametric Cluster-Validity Measurements

Several cluster-validity measures have been proposed in the literature. Most of them compare inter-cluster versus intra-cluster variability and tend to favour configurations with bell-shaped well-separated clusters. In our experiments we considered two non-parametric indices, *isolation* and *connectivity* [7], which can be used for measuring the validity of *arbitrarily shaped* clusters.

Isolation is measured by the k-nearest neighbour norm (NN-norm). In particular, for fixed k (whose specific value is not very critical), the k-nearest neighbour norm $v_k(x)$ of an instance x is defined to be the fraction of the k nearest neighbours of x that have the same cluster label as x. A measure of the homogeneity of the total clustering is computed by averaging over all N points in the data set: $IS = \frac{1}{N} \sum_x v_k(x)$. In our experiments we used $k = 0.01N$.

Connectivity relates to the fact that for any two points in the same cluster, a path should always exist connecting the two points, along which the density of the data remains relatively high. In our implementation, connectivity is quantified as follows: we randomly select K pairs of points (a_i, b_i) $(i = 1, \ldots, K)$, called anchor-points, such that the points of the same pair belong to the same cluster. Then for each pair (a_i, b_i), we consider the middle point $\mu_i = (a_i + b_i)/2$ and compute the local density $f(\mu_i)$ by

convolving the dataset with a unimodal density-kernel of width σ:

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{2\pi\sigma^2} \right)^{d/2} e^{-||x-x_i||^2/2\sigma^2} \tag{5}$$

Then the connectivity measure CN (also called C-norm) is computed as: $CN = \frac{1}{K} \sum_{i=1}^{K} f(\mu_i)$. In our experiments we chose $K = 0.05N$.

A major drawback of the isolation index, is that it does not notice whenever two clusters are merged, even if they are well-separated [7]. In fact, grouping all samples together in one big cluster, will result in an optimal score for this criterion. For this reason connectivity must be considered as a second criterion to penalize solutions that erroneously lump together widely separated clusters. However, since there is a trade-off between connectivity and isolation, the two validity indices should be combined to provide a single cluster-validity index described in Sect. 3.3.

3.2 Experimental Methodology

In order to demonstrate the performance of the Boost-Clustering algorithm we considered both artificial and real-world datasets. In the following, we describe the experimental methodology used to compare the Boost-k-means with the simple k-means algorithm. Note that the same methodology was followed to compare Boost-FCM with simple FCM.

In particular, for each data set and for a specific number of clusters C (the number of clusters for each problem varied from three to six) we applied the following steps:

1. Split the data set into training and testing set of fixed size.
2. Run the simple k-means algorithm 20 times, each time with different initialization, to partition the training set in C clusters. For each of these runs compute the isolation and connectivity value in the testing set.
3. For the 20 runs of the simple k-means compute the average values of isolation (IS_{av}) and connectivity (CN_{av}). Also, the values of isolation (IS_{best}) and connectivity (CN_{best}) indexes are taken concerning the best of the 20 runs of the k-means, ie. the one yielding the smallest clustering error in the training set.
4. Apply the Boost-k-means algorithm with specific values of T and δ on the same training set and compute the values of isolation ($IS_B(T,\delta)$) and connectivity ($CN_B(T,\delta)$) on the test set. Three cases for (T,δ) were considered: $(10,1)$, $(20,1)$, $(10,5)$.

3.3 Combination of Cluster-Validity Measurements

In order to make the two cluster-validity measurements directly comparable, we compute their Z-scores [7]. The Z-score of an observation ξ_i in a sample $\xi_1, \ldots, \xi_l$ is defined as:

$$Z(\xi_i) = \frac{\xi_i - median(\xi)}{MAD(\xi)} \tag{6}$$

where $\xi = \{\xi_1, \ldots, \xi_l\}$ represents the whole sample and MAD stands for *median absolute deviation*:

$$MAD(\xi) = median\{|\xi_i - median(\xi)| : i = 1, \ldots, l\} \tag{7}$$

Now, let us consider $IS = \{IS_{av},\ IS_{best},\ IS_B(20,1),\ IS_B(10,1),\ IS_B(10,5)\}$ be the sample of isolation values and $CN = \{CN_{av},\ CN_{best},\ CN_B(20,1),\ CN_B(10,1),$

$CN_B(10,5)\}$ the sample of connectivity values for the methods we want to compare: 1)Average k-means, 2) Best k-means, 3) Boost-Clustering(10, 1), 4) Boost-Clustering(20,1) and 5) Boost-Clustering(10,5), respectively. Then, the (robust) Z-score for the i-th method is defined as:

$$Z_i = Z(IS\{i\}) + Z(CN\{i\}) \quad i = 1,2,3,4,5 \tag{8}$$

and we consider as the best clustering result the one which maximizes this robust Z-score in the test set.

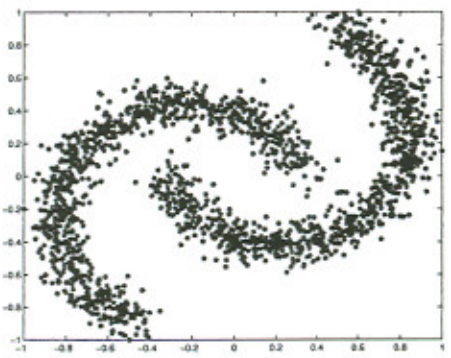

Fig. 1. The Banana data set.

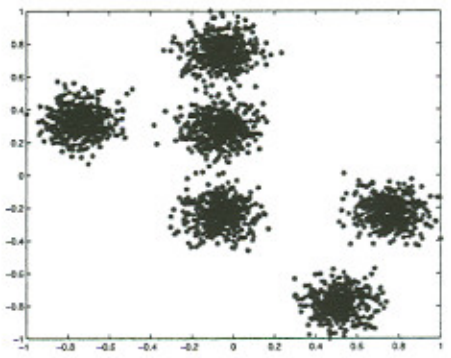

Fig. 2. The Six-Gauss data set.

3.4 Experimental Results and Discussion

Four data sets were used to demonstrate the performance of Boost-Clustering: the Clouds and Phoneme data sets from the ELENA project [8], and the Banana and Six-Gauss which are artificial two-dimensional data sets. The Banana (see Fig. 1) data set consists of two banana shaped clusters while the samples of the Six-Gauss (see Fig. 2) data set have been drawn from a mixture of six Gaussian distributions with unit covariance. All datasets are two-dimensional except for the Phoneme which is five-dimensional.

Table 1 contains the test set experimental results using the simple k-means and the Boost-k-means algorithm. The Z-score (Eq. 8) index is illustrated for each of the five compared cases. In each row in Table 1 the best Z-score (maximum value) is highlighted indicating the best partition result. Similarly, Table 2 contains the test set experimental results comparing the simple fuzzy c-means with the Boost-FCM algorithm.

Running a basic clustering algorithm (such as k-means or FCM) many times (each time with different initialization) one comes up with several partitions. In this case, the best he can do is to get the best clustering solution, e.g. the one yielding the smallest clustering error in the training set. So, it make sense to compare the Boost-clustering result with the best clustering result produced by many applications of the basic clustering algorithm.

The experimental results in Table 1 indicate that Boost-k-means(10,1) gave the best clustering results in 8 out of 16 experiments compared to best k-means. Also,

Table 1. Experimental results for simple k-means and Boost-k-means.

Dataset	Clusters	Best k-means	Average k-means	Boost-k-means(T,δ) (10,1)	(20,1)	(10,5)
Clouds	3	0	0	**2**	-8.711	-40.105
	4	-2.83	0.511	**1.286**	-1	-0.464
	5	0	-1	-1.571	**501**	74.286
	6	0.215	-1	0	**1.09**	0.37
Phoneme	3	**0.549**	0	-0.284	-2.8	-3.468
	4	-0.859	**0.595**	-2	0	-0.643
	5	1	1	**1.431**	-1.832	-3.5
	6	**2**	0.5	-4.492	-3.57	-7.922
Banana	3	-1	0.199	-3.3	-0.592	**0.461**
	4	-0.386	0	**2.409**	1	-31.209
	5	-2.566	-2.024	1	0.2	**7.37**
	6	-1.916	-3.593	0	2.965	**12**
Six-Gauss	3	0	-15.951	-12.183	2.114	**3.5**
	4	-0.1	-48.443	-1	**1**	-24.987
	5	-2	**23.288**	1	-1	-1.749
	6	1	-106	**2**	0	0

Table 2. Experimental results for simple fuzzy C-means and Boost-FCM.

Dataset	Clusters	Best FCM	Average FCM	Boost-FCM(T,δ) (10,1)	(20,1)	(10,5)
Clouds	3	**2**	2	-0.214	-0.25	-2.762
	4	-2.078	-2.078	**2.204**	0	2
	5	-0.534	-0.534	-3	1.534	**31.995**
	6	-3	-3.264	0.299	0	**2.293**
Phoneme	3	-0.94	-0.94	-1	-1.5	**1**
	4	0	-0.005	**3.155**	0	1.715
	5	-0.868	-0.868	-3.975	**2.965**	0.702
	6	-2.558	-2.558	**5.837**	0.439	-1.441
Banana	3	0	-0.25	**2.597**	2.458	-0.02
	4	-3.811	-3.811	1.199	-2.2	**12.947**
	5	-1	-1.01	2.332	-1.5	**5.481**
	6	-1	-1.5	**1.379**	-1.804	-4.783
Six-Gauss	3	0.542	0.542	**2**	-0.542	-49.654
	4	-2	-7.955	-1.5	1.354	**3.175**
	5	-3	5.667	**22.644**	1.828	2
	6	2	-1	**2**	2	1

Boost-k-means(20,1) outperforms the best k-means in 11 over 16 cases and Boost-k-means(10,5) in 9 cases. Similarly, the experimental results in Table 2 indicate that Boost-FCM(10,1) outperforms the best FCM in 12 out of 16 experiments. Also, both Boost-FCM(20,1) and Boost-FCM(10,5) algorithms outperform the best FCM in 11 cases.

One important conclusion that can be drawn is that in most cases the Boost-clustering alogirthm provides better partition results (in 27 of total 32 experiments) than a simple clustering algorithm. This strongly indicates that boosting a basic clustering algorithm for a small number of iterations can give better results than running many times the basic clustering algorithm and selecting the partition of best run according to the minimum value of the objective function.

Also, the obtained results show that in the 23 out of 27 cases that Boost-clustering gave the best partition result, only $T = 10$ iterations were used. The Boost-clustering performance degradation that sometimes occurs by increasing the number of iterations is in accordance with analogous results reported in the literature concerning the AdaBoost method and its variants when applied to classification problems [6, 9]. In the case of the Boost-clustering algorithm, overtraining may be observed after many iterations, especially for data sets with noisy patterns or outliers which are hard to be clustered. Another reason for the Boost-clustering performance degradation after many iterations is the distortion of the overall structure of the original data set (especially for small data sets) due to the resampling, so an early stopping criterion for Boost-clustering is critical.

There exist several directions for future work with the boost-clustering algorithm. The most important direction deals with the development of criteria for early stopping (specification of the optimal number of iterations T). Another interesting issue is the specification of alternative ways for the evaluation of how well a data point has been clustered. Finally, a more thorough experimental evaluation of the method is needed using many high-dimensional datasets as well as real problems (eg. image segmentation) and considering several types of basic clustering algorithms.

References

[1] Frossyniotis D., Pertselakis M., and Stafylopatis A. A Multi-Clustering Fusion Algorithm. In *Proceedings of the Second Hellenic Conference on Artificial Intelligence (SETN2002)*, LNAI 2308, pages 225–236, Thessaloniki, Greece, April 11-12 2002. Springer-Verlag.

[2] Fred A. Finding Consistent Clusters in Data Partitions. In *Proceedings of the Second International Workshop on Multiple Classifier Systems (MCS 2001)*, LNCS 2096, pages 309–318, Cambridge, UK, July 2-4 2001. Springer.

[3] Smyth P. Clustering Using Monte Carlo Cross-Validation. In *Proceedings Knowledge Discovery and Data Mining*, pages 126–133, 1996.

[4] Fisher D.H. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[5] Halkidi M., Batistakis Y., and Vazirgiannis M. Clustering algorithms and validity measures. In *Proceedings of the SSDBM conference*, Virginia,USA, July 2001.

[6] Freund Y. and Schapire R. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, Bari, Italy, 1996.

[7] Pauwels E. and Frederix G. Finding salient regions in images: Nonparametric clustering for image segmentation and grouping. *Computer Vision and Image Understanding*, 75:73–85, 1999.

[8] ESPRIT Basic Research Project ELENA (no. 6891). [ftp://ftp.dice.ucl.ac.be/pub/neural-nets/ELENA/databases], 1995.

[9] Wickramaratna J., Holden S., and Buxton B. Performance degradation in boosting. In *Proceedings of the 2nd International Workshop on Multiple Classifier Systems MCS2001*, volume 2096, pages 11–21. Springer, 2001.